



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

ESCOLA DE INFORMÁTICA APLICADA

## ESPECIFICAÇÃO DE REQUISITOS

PABLO MARTINS BORGES

JOÃO PEDRO RIBEIRO

RENAN BARBOSA DE LIMA

LEONARDO DE LIMA GONÇALVES

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2024

<b>Introdução</b>	<b>3</b>
<b>Termos técnicos</b>	<b>3</b>
<b>Visão Geral do produto</b>	<b>3</b>
Descrição do Minimundo	3
Delimitação do escopo inicial	4
<b>Atores/Usuários e Envolvidos</b>	<b>4</b>
<b>Requisitos</b>	<b>5</b>
Regras de negócio	5
Requisitos Funcionais	6
Requisitos Não-Funcionais	7
<b>Diagrama de Caso de uso</b>	<b>8</b>
<b>Casos de uso</b>	<b>9</b>
Caso de uso 01	9
Caso de uso 02	13
Caso de uso 03	14
Caso de uso 04	15
Caso de uso 05	18
Caso de uso 06	19
Caso de uso 07	20
Caso de uso 08	21
Caso de uso 09	22
Caso de uso 10	23
Caso de uso 11	24
<b>Diagrama de Classe</b>	<b>25</b>
<b>Esquema Lógico do Banco de Dados Relacional</b>	<b>26</b>
<b>Diagrama de alocação de componentes</b>	<b>27</b>
<b>Diagrama de Estado do CDU 05</b>	<b>29</b>
<b>Diagrama de Sequência do CDU 7</b>	<b>29</b>
<b>Esquema Físico do Banco de Dados Relacional/DDL</b>	<b>30</b>
Script	30
<b>Carga inicial do Banco de Dados Relacional/DML</b>	<b>31</b>
<b>Esquema Físico e Carga Inicial do Banco de Dados Não-Convencional</b>	<b>32</b>
<b>Diagrama do Banco de Dados Não-Convencional</b>	<b>33</b>

# Introdução

O intuito deste documento é apresentar e explorar a plataforma GaMeA\_list. Este software possui como principais objetivos para com seus usuários: proporcionar um *backlog* de jogos, registrar seu progresso de seus jogos atuais, permitir que haja interação com outros usuários no que tange aos jogos.

Com isso, este documento contém a definição de: termos técnicos, minimundo, delimitação de escopo, atores envolvidos, os requisitos do sistema e casos de uso.

## Termos técnicos

- Platinar: obter todas as conquistas de um título;
- Dropar: começar um jogo e parar de jogá-lo;
- Finalizar: jogar de forma que se alcance os créditos.
- Streaming: modelo de aluguel de jogos;
- Biblioteca: lista de jogos possuídos.

## Visão Geral do produto

### Descrição do Minimundo

Existem atualmente vários aplicativos/sites que listam e registram jogos, assim como as horas que um usuário os jogou, mas esses sistemas ou são plataformas de venda de licenças de jogos, as quais registram apenas os jogos que são comercializados pela plataforma, assim como apenas as métricas de interação dos usuários com esses jogos, ou são apenas base de dados de jogos já lançados, sem a possibilidade de se ter usuários e as métricas de suas interações com os jogos. Outra questão é que com as plataformas gigantes neste mercado o número de jogos obtidos pode sair do controle rapidamente, seja pelas promoções com descontos altíssimos(*Steam*), dados semanalmente(*Epic Game Store*) ou disponibilizados via *streaming* (*Xbox Game Pass*).

Diante desta problemática surge o GaMeA\_list, que concentra as informações relativas aos jogos de um determinado usuário de forma centralizada e organizada. Para evitar compras desnecessárias e dar feedback sobre os jogos.

## Delimitação do escopo inicial

O projeto inicial contemplará a criação do software GaMeA\_list, o qual num contexto de jogos funciona como uma biblioteca, base de dados. E para estes últimos com a possibilidade de adicionar as seguintes métricas: horas jogadas, nota, se o jogo foi platinado, finalizado, está sendo jogado ou dropado. O usuário poderá registrar também uma breve análise sobre o jogo.

A aplicação serve também como uma rede social na qual um usuário pode ver as listas de outros usuários, assim como suas métricas e mudanças feitas nos jogos. Além de uma base de dados sobre jogos existentes, a qual é alimentada pelos próprios usuários quando registram um jogo que ainda não exista no banco de dados do produto.

Não será contemplado no escopo deste projeto, inicialmente, o redirecionamento para lojas que vendem os jogos registrados na plataforma, assim como funções de moderação dos usuários para a adição de jogos à base de dados da plataforma ou de comentários feitos.

## Atores/Usuários e Envolvidos

Nome	Responsabilidades	Perfil
<b>Usuário autenticado</b>	<ul style="list-style-type: none"><li>● Manter seu perfil;</li><li>● Manter jogos;</li><li>● Administrar suas listas;</li><li>● Interagir com atividades de outros usuários;</li></ul>	<i>Utilizador da plataforma.</i>
<b>Usuário não autenticado</b>	<ul style="list-style-type: none"><li>● Visualizar os jogos cadastrados.</li></ul>	<i>Utilizador limitado da plataforma.</i>

## Requisitos

## Regras de negócio

ID	Descrição
<b>RN01</b>	Existem 4 estados fixos para os jogos em bibliotecas (não jogado, finalizados, dropados e finalizados).
<b>RN02</b>	Um jogo pode ser adicionado por vários usuários.
<b>RN03</b>	Um cadastro de um usuário deve conter: email, senha, nome e nome de usuário.
<b>RN04</b>	Uma inserção de um jogo na biblioteca deve conter a plataforma na qual se possui o jogo.
<b>RN05</b>	Um cadastro de um jogo deve conter: título, imagem, publisher e estúdio
<b>RN06</b>	Uma inserção de um jogo como finalizado, platinado ou dropado deve conter: horas jogadas, curtiu o jogo e análise.
<b>RN07</b>	Uma consulta a outro usuário não deve aparecer email, senha e nome.
<b>RN08</b>	Um usuário pode adicionar vários jogos às bibliotecas.
<b>RN09</b>	Um usuário pode seguir n usuários.

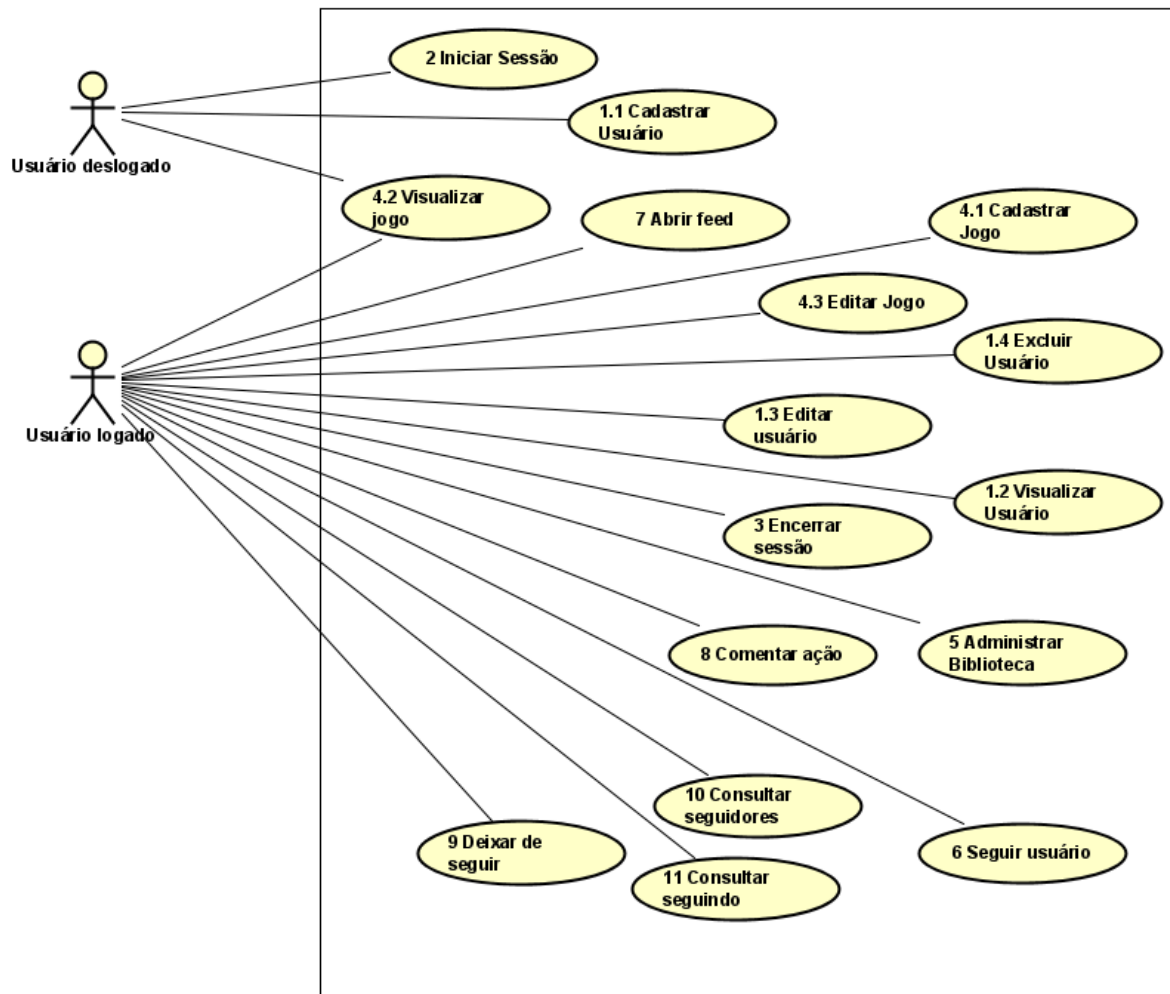
## Requisitos Funcionais

ID	Descrição
<b>RF01</b>	O sistema deve permitir ao usuário se autenticar.
<b>RF02</b>	O sistema deve permitir ao usuário autenticado encerrar a sessão.
<b>RF03</b>	O sistema deve permitir que o usuário mantenha seu próprio perfil.
<b>RF04</b>	A plataforma deve permitir ao usuário autenticado administrar sua biblioteca de jogos.
<b>RF05</b>	O sistema deve permitir consulta aos jogos por todos os usuários.
<b>RF06</b>	O sistema deve permitir que um usuário autenticado mantenha um jogo.
<b>RF07</b>	O sistema deve permitir que um usuário autenticado siga ou deixe de seguir outro usuário.
<b>RF08</b>	O sistema deve mostrar ações relacionadas aos jogos em biblioteca feitas por usuários seguidos em formato de feed.
<b>RF09</b>	Ao inserir um jogo na biblioteca o sistema deve pedir a plataforma que o usuário autenticado possui o jogo.
<b>RF10</b>	Ao inserir ou modificar um jogo com o estado de “dropado”, “finalizado” ou “platinado” o sistema deve pedir horas jogadas maiores que 0, nota do jogo e uma breve análise.
<b>RF11</b>	O sistema só deve permitir a mudança de status de um jogo na biblioteca do usuário para “jogando” se as horas jogadas forem maiores que 0 horas.
<b>RF12</b>	O sistema deve permitir que um usuário autenticado comente as ações que aparecem no seu feed.
<b>RF13</b>	O sistema deve publicar as ações que um usuário faz relacionadas à adição/alteração de jogo em biblioteca.
<b>RF14</b>	O sistema deve permitir que um usuário autenticado consulte os seguidores ou seguindo de outro usuário.

## Requisitos Não-Funcionais

ID	Descrição
<b>RNF1</b>	O sistema deve ser compatível com o sistema operacional Android 11.0.
<b>RNF2</b>	O sistema deve cumprir com os requisitos referentes à política de privacidade da Play Store.
<b>RNF3</b>	O sistema deve possuir uma ótima usabilidade quanto a apresentação, sendo o mais intuitivo possível.
<b>RNF4</b>	O sistema deve possuir uma ótima manutenibilidade quanto a modificabilidade, permitindo alterações com facilidade e rapidez
<b>RNF5</b>	O sistema deve garantir a acurácia de suas funcionalidades, executando suas funções com precisão e resultados esperados
<b>RNF6</b>	O sistema deve possuir total segurança e todos os dados fornecidos pelos usuários serão protegidos no banco de dados
<b>RNF7</b>	O sistema deve possuir uma ótima operacionalidade, na qual sua interface não possuirá redundância ou impor qualquer tipo de dificuldade no seu uso
<b>RNF7</b>	Um usuário deve estar autenticado para poder acessar o sistema totalmente.
<b>RNF8</b>	Uma consulta aos jogos não depende de autenticação no sistema.

# Diagrama de Caso de uso





# Casos de uso

## Caso de uso 01

CDU01.1	
Nome	Cadastrar Usuário
Objetivo	Permitir o cadastro de informações essenciais para que o usuário possa realizar autenticação
Ator	Usuário
Pré-condições	Usuário não cadastrado no sistema
Trigger	Ator pressiona o botão “inscrever-se” na tela inicial
Fluxo Principal	<ol style="list-style-type: none"><li>1. O sistema exibe a tela de inserção de dados.</li><li>2. O ator insere os dados necessários para cadastro e confirma o cadastro.[E1]</li><li>3. O sistema valida, salva os dados e envia um email para confirmação. [E2][E3]</li><li>4. O ator clica no link enviado ao email cadastrado.</li><li>5. O sistema confirma a ativação de conta.</li></ol>
Fluxo Alternativo	N/A
Fluxo de Exceção	<p>E1 - Desistência A1.1 - O ator sai da tela de cadastro sem concluir a operação.</p> <p>E2 - Dados Inválidos E2.1 - O sistema informa que algum dos dados informados está inválido ou nulo E2.2 - Volta para o passo 1.</p> <p>E3 - Email já cadastrado E3.1 - O sistema informa que já existe um usuário com o email cadastrado no sistema. E3.2 - Volta para o passo 1.</p>
Extensões	N/A

<b>Pós-condições</b>	Usuário apto a fazer login no sistema.
<b>Regras de negócio</b>	RN03
<b>Requisitos Funcionais</b>	RF03 O sistema deve permitir que o usuário mantenha seu próprio perfil.
<b>Tipo de UC</b>	Primário

CDU01.2	
<b>Nome</b>	Visualizar usuário
<b>Objetivo</b>	Permitir a visualização de informações de um usuário
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ator autenticado no sistema
<b>Trigger</b>	Ator seleciona a opção de ver dados do Usuário
<b>Fluxo Principal</b>	1. O sistema exibe os dados do usuário autenticado e suas ações. [A1]
<b>Fluxo Alternativo</b>	A1 - Ator visualiza perfil de outro usuário A1.1 - O sistema mostrará apenas o nome de usuário e suas ações.
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	As informações do usuário são exibidas
<b>Regras de negócio</b>	RN07
<b>Requisitos Funcionais</b>	RF03 O sistema deve permitir que o usuário mantenha seu próprio perfil.
<b>Tipo de UC</b>	Primário

CDU01.3	
<b>Nome</b>	Editar usuário
<b>Objetivo</b>	Permitir a edição de informações de cadastro

<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ator autenticado no sistema
<b>Trigger</b>	Ator seleciona o botão “Alterar perfil”
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema exibe a tela de alteração de dados.</li> <li>2. O ator insere os dados necessários para atualização e confirma o cadastro.[E1]</li> <li>3. O sistema valida e salva os dados [E2][E3]</li> </ol>
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	<p>E1 - Desistência E1.1 - O ator sai da tela de atualização sem concluir a operação.</p> <p>E2 - Dados Inválidos E2.1 - O sistema informa que algum dos dados informados está inválido ou nulo E2.2 - Volta para o passo 1.</p> <p>E3 - Email já cadastrado E3.1 - O sistema informa que já existe um usuário com o email cadastrado no sistema. E3.2 - Volta para o passo 1.</p>
<b>Extensões</b>	N/A
<b>Pós-condições</b>	Os dados do usuário são atualizados.
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF03 O sistema deve permitir que o usuário mantenha seu próprio perfil.
<b>Tipo de UC</b>	Primário

CDU01.4	
<b>Nome</b>	Excluir usuário
<b>Objetivo</b>	Permitir a exclusão do perfil de um usuário.
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ator autenticado no sistema.

<b>Trigger</b>	Ator seleciona “excluir perfil”.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema verifica se o ator quer realmente excluir sua conta.</li> <li>2. O ator confirma a exclusão.[E1]</li> <li>3. O sistema realiza a exclusão</li> </ol>
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	<p>E1 - Desistência</p> <p>E1.1 - O ator cancela a exclusão.</p>
<b>Extensões</b>	N/A
<b>Pós-condições</b>	<p>O usuário é removido do sistema.</p> <p>O usuário não possui mais acesso ao sistema.</p>
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF03 O sistema deve permitir que o usuário mantenha seu próprio perfil.
<b>Tipo de UC</b>	Primário

## Caso de uso 02

CDU02	
<b>Nome</b>	Iniciar sessão
<b>Objetivo</b>	iniciar uma sessão na plataforma
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ator não autenticado
<b>Trigger</b>	Ator aperta “autenticar”
<b>Fluxo Principal</b>	<ol style="list-style-type: none"><li>1. Sistema exibe tela de acesso com os campos “Email” e “Senha” para serem preenchidos</li><li>2. Ator digita seu email e senha e seleciona o botão “Entrar”</li><li>3. Sistema valida e-mail e senha, autentica ator e permite o início de uma sessão [E1]</li></ol>
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	E1 - Sistema não consegue validar <ol style="list-style-type: none"><li>1. Sistema exibe mensagem alertando que algum dos dados está inválido.</li></ol>
<b>Extensões</b>	N/A
<b>Pós-condições</b>	Ator autenticado
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF01 O sistema deve permitir ao usuário se autenticar.
<b>Tipo de UC</b>	Primário

## Caso de uso 03

CDU03	
Nome	Encerrar sessão
Objetivo	Encerrar uma sessão na plataforma
Ator	Usuário
Pré-condições	Ator autenticado
Trigger	Ator seleciona o botão "encerrar sessão" na tela principal
Fluxo Principal	1. Sistema encerra a sessão do usuário
Fluxo Alternativo	N/A
Fluxo de Exceção	N/A
Extensões	N/A
Pós-condições	Ator na tela de acesso
Regras de negócio	N/A
Requisitos Funcionais	RF02 O sistema deve permitir ao usuário autenticado encerrar a sessão.
Tipo de UC	Primário

## Caso de uso 04

CDU04.1	
<b>Nome</b>	Cadastrar Jogo
<b>Objetivo</b>	Permitir o cadastro de informações essenciais para que os jogos sejam adicionados às bibliotecas.
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Jogo não cadastrado no sistema
<b>Trigger</b>	Ator pressiona o botão “adicionar jogo”
<b>Fluxo Principal</b>	<ol style="list-style-type: none"><li>1. O sistema exibe a tela de inserção de dados.</li><li>2. O ator insere os dados necessários para cadastro e confirma o cadastro.[E1]</li><li>3. O sistema valida e salva os dados [E2][E3]</li></ol>
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	<p>E1 - Desistência E1.1 - O ator sai da tela de cadastro sem concluir a operação.</p> <p>E2 - Dados Inválidos E2.1 - O sistema informa que algum dos dados informados está inválido ou nulo E2.2 - Volta para o passo 1.</p> <p>E3 - Jogo já cadastrado E3.1 - O sistema informa que já existe um jogo com esse título no sistema.</p>
<b>Extensões</b>	N/A
<b>Pós-condições</b>	Jogo disponível para ser inserido nas bibliotecas.
<b>Regras de negócio</b>	RN05
<b>Requisitos Funcionais</b>	RF06 O sistema deve permitir que um usuário autenticado mantenha um jogo.
<b>Tipo de UC</b>	Primário

CDU04.2	
<b>Nome</b>	Visualizar jogo
<b>Objetivo</b>	Permitir a visualização de informações de cadastro de um jogo
<b>Ator</b>	Usuário autenticado ou Usuário não autenticado
<b>Pré-condições</b>	N/A
<b>Trigger</b>	Ator aperta “detalhes”
<b>Fluxo Principal</b>	1. O sistema exibe os dados do jogo.
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	As informações do jogo são exibidas
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF06 O sistema deve permitir que um usuário autenticado mantenha um jogo.
<b>Tipo de UC</b>	Primário

CDU04.3	
<b>Nome</b>	Editar jogo
<b>Objetivo</b>	Permitir a edição de informações de cadastro de um jogo
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator autenticado no sistema
<b>Trigger</b>	Ator seleciona o botão “Alterar jogo”
<b>Fluxo Principal</b>	1. O sistema exibe a tela de alteração de dados. 2. O ator insere os dados necessários para atualização e confirma o cadastro.[E1]



	3. O sistema valida e salva os dados [E2][E3]
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	<p>E1 - Desistência E1.1 - O ator sai da tela de atualização sem concluir a operação.</p> <p>E2 - Dados Inválidos E2.1 - O sistema informa que algum dos dados informados está inválido ou nulo E2.2 - Volta para o passo 1.</p> <p>E3 - Jogo já cadastrado E3.1 - O sistema informa que já existe um jogo com esse título no sistema.</p>
<b>Extensões</b>	N/A
<b>Pós-condições</b>	Informações do jogo atualizadas.
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF06 O sistema deve permitir que um usuário autenticado mantenha um jogo.
<b>Tipo de UC</b>	Primário

## Caso de uso 05

CDU05	
<b>Nome</b>	Administrar biblioteca
<b>Objetivo</b>	Adicionar ou alterar jogo numa biblioteca
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator autenticado
<b>Trigger</b>	O ator pressiona o botão de “adicionar jogo” ou “alterar jogo”
<b>Fluxo Principal</b>	<ol style="list-style-type: none"><li>1. Ator escolhe um jogo cadastrado no sistema.</li><li>2. O sistema mostra as opções de estado.[A1]</li><li>3. O ator escolhe o estado. [E1]</li><li>4. O sistema mostra os campos necessários de acordo com o estado.</li><li>5. O ator coloca os dados necessários e confirma.</li><li>6. O sistema insere o jogo desejado na biblioteca e faz uma publicação.</li></ol>
<b>Fluxo Alternativo</b>	<p>A1 - Modificação de estado</p> <p>A1.1 - O sistema mostra as opções de estado.[A1]</p> <p>A1.2 - O ator escolhe o estado. [E1]</p> <p>A1.3 - O sistema mostra os campos necessários de acordo com o estado.</p> <p>A1.4 - O ator coloca os dados necessários e confirma.</p> <p>A1.5 - O sistema atualiza o jogo na biblioteca e faz uma publicação.</p>
<b>Fluxo de Exceção</b>	<p>E1 - Desistência</p> <p>E1.1 - O ator não finaliza a inclusão.</p>
<b>Extensões</b>	N/A
<b>Pós-condições</b>	O jogo está inserido na biblioteca desejado e pode ser visualizado.
<b>Regras de negócio</b>	RN04, RN06,RN08
<b>Requisitos Funcionais</b>	<p>RF04 A plataforma deve permitir ao usuário autenticado administre sua biblioteca de jogos..</p> <p>RF09 Ao inserir um jogo na biblioteca o sistema deve pedir a plataforma que o usuário autenticado possui o jogo.</p>

	<p>RF10 - Ao inserir ou modificar um jogo com o estado de “dropado”, “finalizado” ou “platinado” o sistema deve pedir horas jogadas maiores que 0, nota do jogo e uma breve análise.</p> <p>RF11 - O sistema só deve permitir a mudança de status de um jogo na biblioteca do usuário para “jogando” se as horas jogadas forem maiores que 0 horas.</p> <p>RF13 - O sistema deve publicar as ações que um usuário faz relacionadas à adição/alteração de jogo em biblioteca.</p>
<b>Tipo de UC</b>	Primário

## Caso de uso 06

CDU06	
<b>Nome</b>	Seguir usuário
<b>Objetivo</b>	Seguir um usuário da plataforma
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator autenticado e não segue o usuário alvo.
<b>Trigger</b>	Usuário aperta “seguir”
<b>Fluxo Principal</b>	1. O sistema registra que o ator está seguindo o usuário.
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	O ator receberá em seu feed ações do usuário seguido.
<b>Regras de negócio</b>	RN09
<b>Requisitos Funcionais</b>	RF07 O sistema deve permitir que um usuário autenticado siga ou deixe de seguir outro usuário.
<b>Tipo de UC</b>	Primário

## Caso de uso 07

CDU07	
<b>Nome</b>	Abrir feed
<b>Objetivo</b>	Ver as ações dos outros usuários
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator segue outros usuários
<b>Trigger</b>	Ator apertou o botão do feed.
<b>Fluxo Principal</b>	1. As ações dos usuários seguidos aparecem em ordem de publicação da mais recente para a mais antiga. Cada ação pode ser comentada.
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	O ator pode interagir com as publicações dos usuários seguidos.
<b>Regras de negócio</b>	RN09
<b>Requisitos Funcionais</b>	RF08 O sistema deve mostrar ações relacionadas aos jogos em biblioteca feitas por usuários seguidos em formato de feed.
<b>Tipo de UC</b>	Primário

## Caso de uso 08

CDU08	
Nome	Comentar ação
Objetivo	Interagir com ação
Ator	Usuário autenticado
Pré-condições	Ator segue outros usuários e tem ações em seu feed
Trigger	Ator pressionou o input de comentário.
Fluxo Principal	<ol style="list-style-type: none"><li>1. O ator faz seu comentário e confirma. [E1]</li><li>2. O sistema registra o comentário</li></ol>
Fluxo Alternativo	N/A
Fluxo de Exceção	<p>E1 - Desistência</p> <p>E1.1 - O ator não realiza o comentário.</p>
Extensões	N/A
Pós-condições	O comentário está público na ação.
Regras de negócio	N/A
Requisitos Funcionais	RF12 O sistema deve permitir que um usuário autenticado comente as ações que aparecem no seu feed.
Tipo de UC	Primário

## Caso de uso 09

CDU09	
<b>Nome</b>	Deixa de seguir
<b>Objetivo</b>	Deixar de seguir um usuário
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator segue usuário.
<b>Trigger</b>	Ator pressionou o botão de “deixar de seguir”
<b>Fluxo Principal</b>	1. O sistema registra que parou de seguir.
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	O usuário não aparecerá mais no feed do ator
<b>Regras de negócio</b>	RN09
<b>Requisitos Funcionais</b>	RF07 O sistema deve permitir que um usuário autenticado siga ou deixe de seguir outro usuário.
<b>Tipo de UC</b>	Primário

## Caso de uso 10

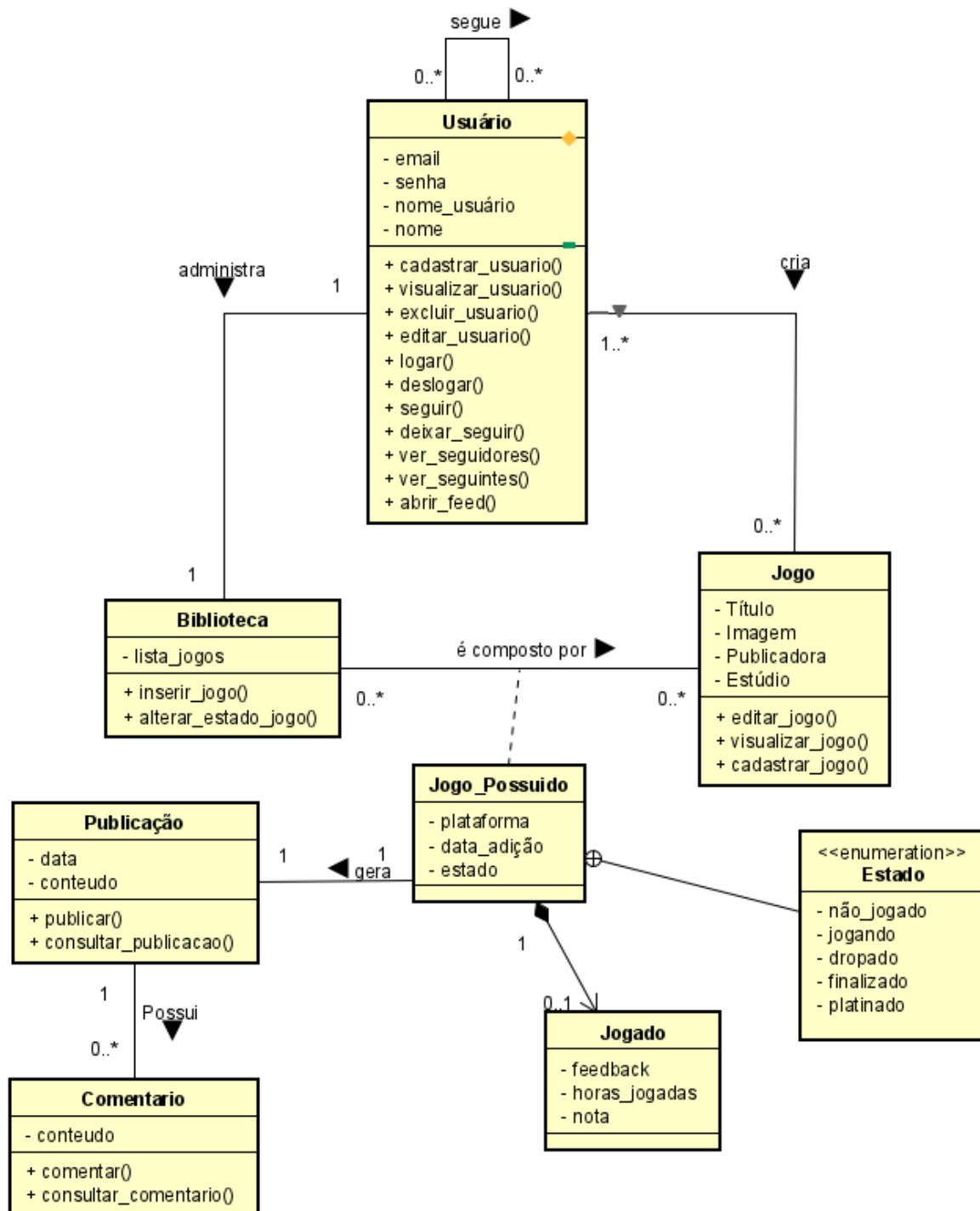
CDU10	
<b>Nome</b>	Consultar seguidores
<b>Objetivo</b>	Listar todos os seguidores de um usuário
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator autenticado.
<b>Trigger</b>	Ator pressionou o botão de “consultar seguidores”
<b>Fluxo Principal</b>	1. O sistema exibe todos os seguidores do usuário.
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	Ator observa os seguidores de um usuário.
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF14 O sistema deve permitir que um usuário autenticado consulte os seguidores ou seguintes de outro usuário.
<b>Tipo de UC</b>	Primário

## Caso de uso 11

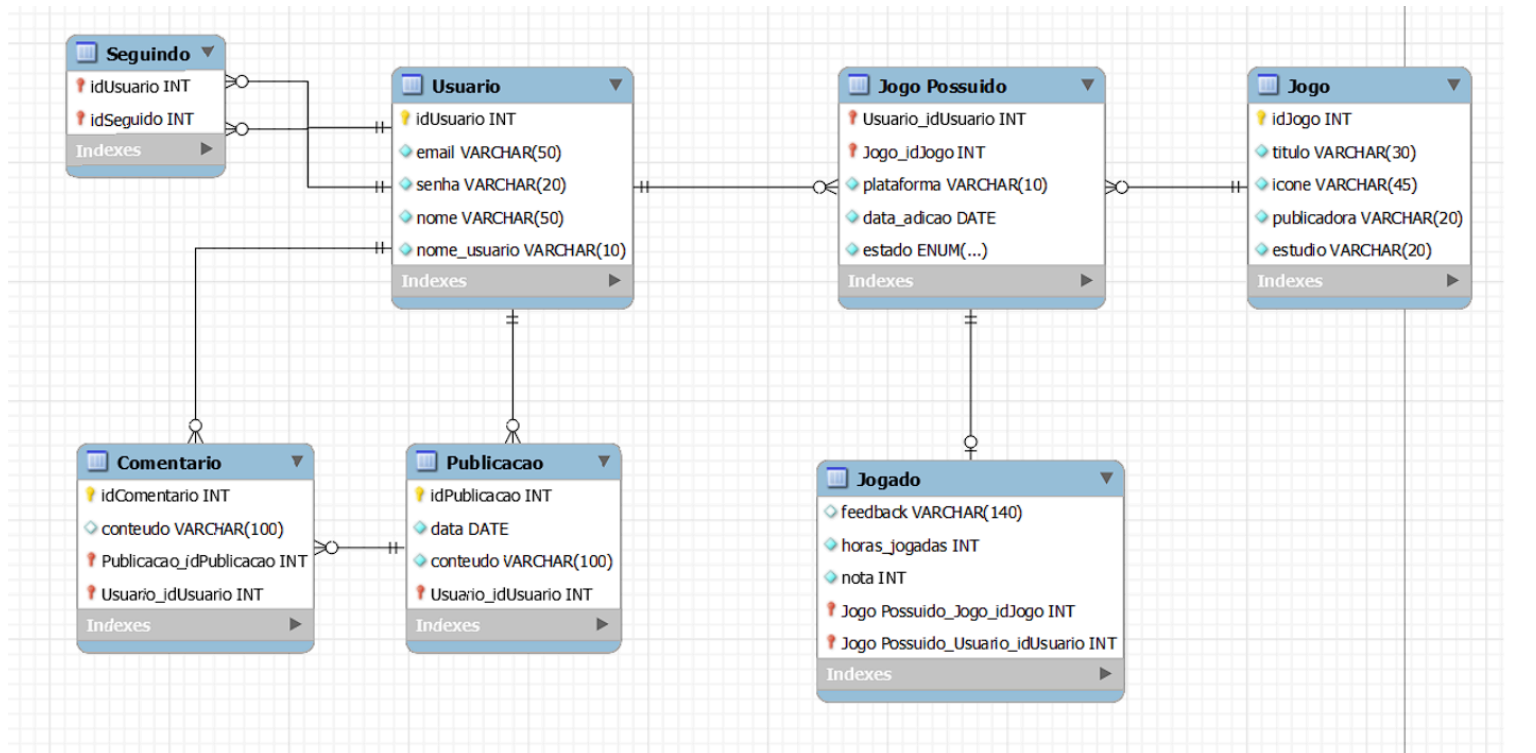
CDU11	
<b>Nome</b>	Consultar seguindo
<b>Objetivo</b>	Listar todos os usuários seguidos pelo usuário alvo
<b>Ator</b>	Usuário autenticado
<b>Pré-condições</b>	Ator autenticado.
<b>Trigger</b>	Ator pressionou o botão de “consultar seguidores”
<b>Fluxo Principal</b>	1. O sistema exibe todos os usuários seguidos pelo usuário alvo.
<b>Fluxo Alternativo</b>	N/A
<b>Fluxo de Exceção</b>	N/A
<b>Extensões</b>	N/A
<b>Pós-condições</b>	Ator observa os usuários seguidos pelo usuário alvo
<b>Regras de negócio</b>	N/A
<b>Requisitos Funcionais</b>	RF14 O sistema deve permitir que um usuário autenticado consulte os seguidores ou seguindo de outro usuário.
<b>Tipo de UC</b>	Primário



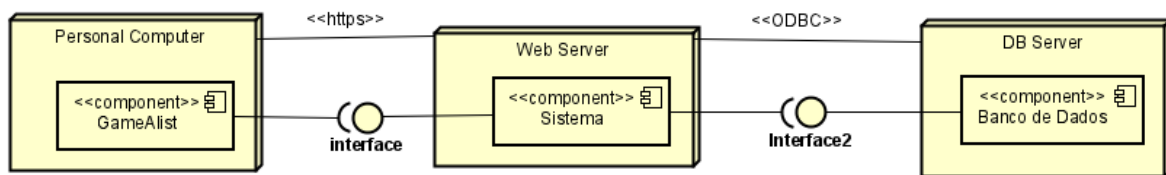
# Diagrama de Classe



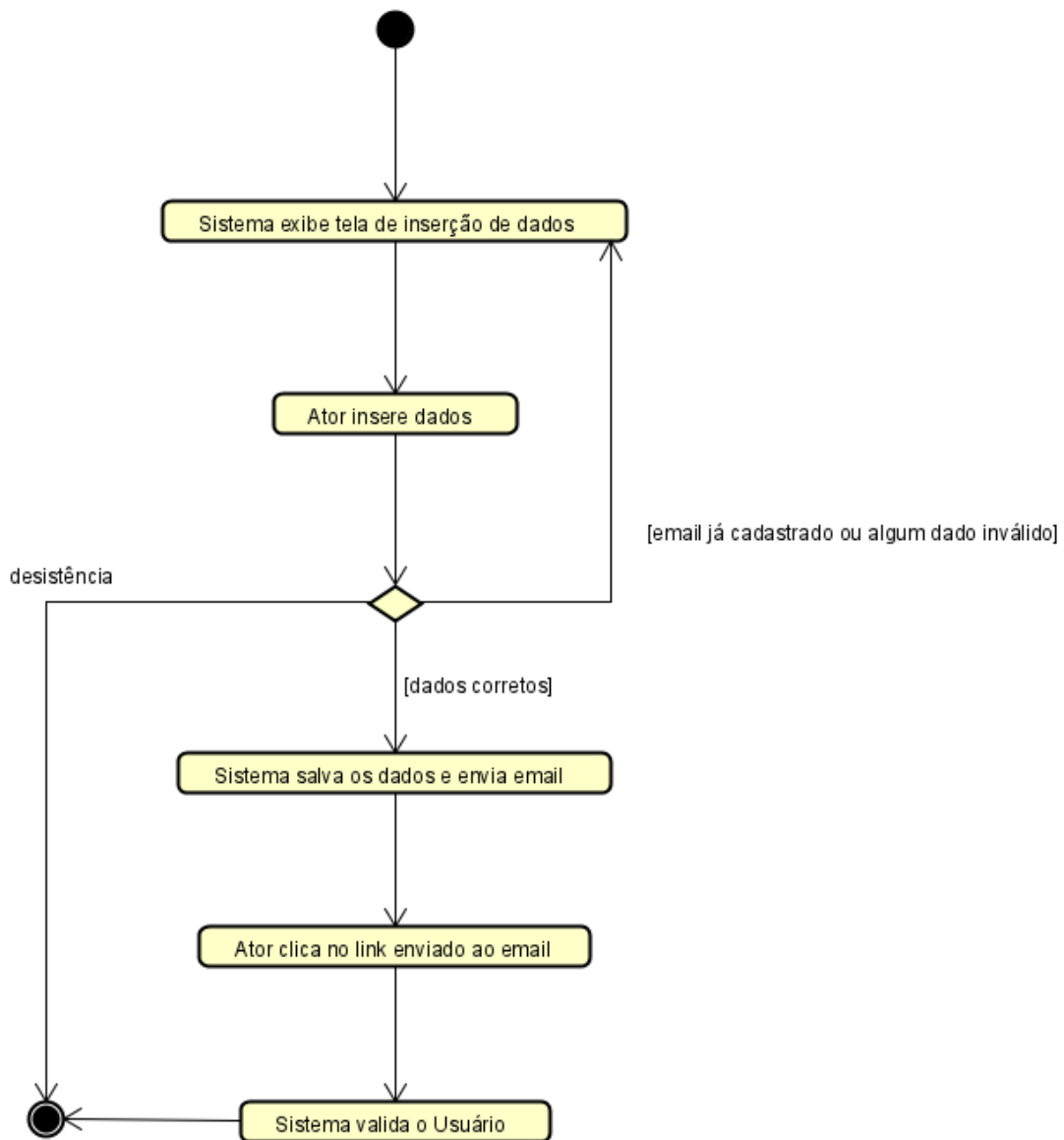
# Esquema Lógico do Banco de Dados Relacional



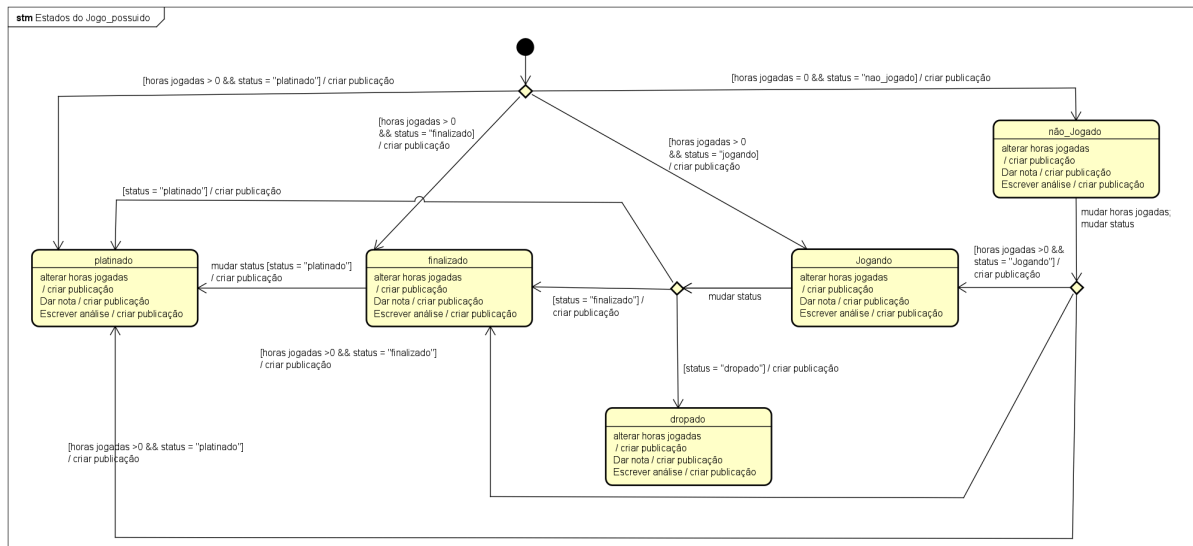
# Diagrama de alocação de componentes



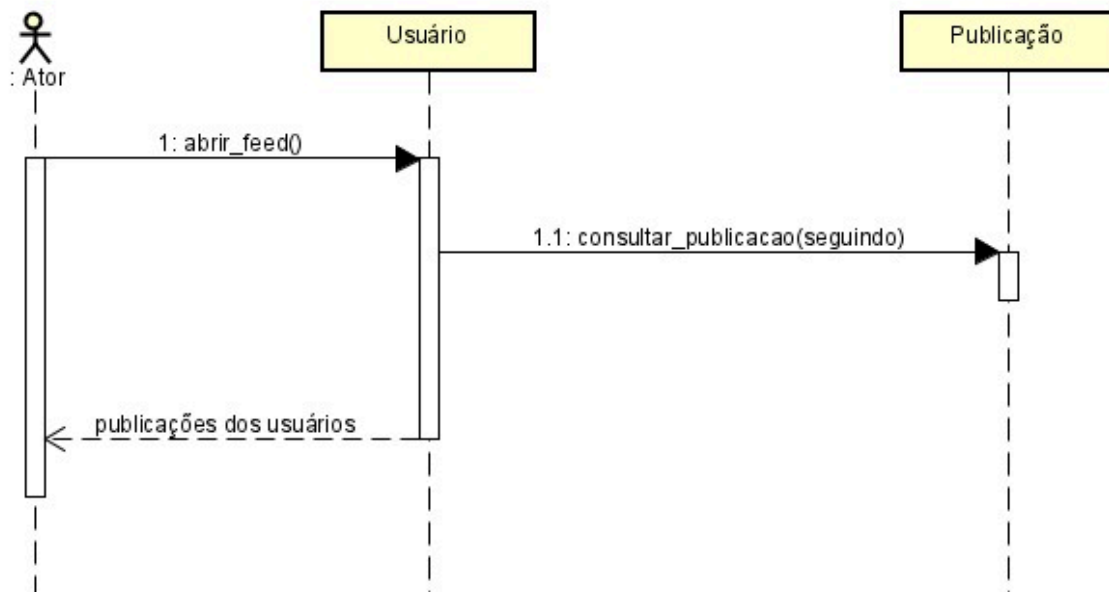
# Diagrama de Atividade do CDU 1.1



# Diagrama de Estado do CDU 05



# Diagrama de Sequência do CDU 7



# Esquema Físico do Banco de Dados Relacional/DDL

Banco de dados: MySQL Community 8.0.37

## Script

```
create table if not exists jogo (  
    idJogo int primary key auto_increment not null,  
    titulo varchar(30) not null,  
    icone varchar(50) not null,  
    publicadora varchar(20) not null,  
    estúdio varchar(20) not null  
);  
  
create table if not exists usuario (  
    idUsuario int primary key auto_increment not null,  
    email varchar(50) not null unique,  
    senha varchar(20) not null,  
    nome varchar(50) not null,  
    nome_usuario varchar(20) not null unique  
);  
  
create table if not exists seguindo (  
    idUsuario int not null,  
    idSeguido int not null,  
    primary key (idUsuario, idSeguido),  
    foreign key (idUsuario) references usuario(idUsuario),  
    foreign key (idSeguido) references usuario(idUsuario)  
);  
  
create table if not exists publicacao(  
    idPublicacao int primary key auto_increment not null,  
    dataPublicacao DATETIME not null,  
    conteudo varchar (200) not null,  
    idUsuario int not null,  
    foreign key (idUsuario) references usuario(idUsuario),  
    index idUsuario(idUsuario)  
);  
  
create table if not exists comentario(  
    idComentario int primary key auto_increment not null,  
    dataComentario DATETIME not null,  
    conteudo varchar (100),  
    idUsuario int not null,
```

```
idPublicacao int not null,  
foreign key (idUserario) references usuario(idUsuario),  
foreign key (idPublicacao) references publicacao(idPublicacao),  
index idPublicacao(idPublicacao)  
);
```

## Carga inicial do Banco de Dados Relacional/DML

```
insert into usuario (idUserario,email,senha,nome,nome_usuario) VALUES  
( 'usuario@email.com', 'senha1234', 'Usuario 1', 'UserOne');
```

```
insert into usuario (idUserario ,email,senha,nome,nome_usuario) VALUES  
( 'noobmaster@gmail.com', 'senha1234', 'Joao', 'NoobMaster69');
```

```
insert into usuario (idUserario,email,senha,nome,nome_usuario) VALUES  
( 'usuario2@email.com', 'senha5678', 'Usuario 2', 'UserTwo');
```

```
insert into seguindo values (1,2);  
insert into seguindo values (1,3);
```

```
insert into publicacao(dataPublicacao, conteudo, idUsuario) values (NOW(), "NoobMaster69  
finalizou 'Elden Ring' com 47 horas", 2);
```

```
insert into comentario(dataComentario, conteudo, idUsuario, idPublicacao) values  
(NOW(), 'Legal esse ai é brabo', 3, 1);
```

```
insert into jogo ( idJogo,titulo,icone,,publicadora,estudio) values (11, 'Counter Strike  
2', 'url_ImagemCounterStrike', 'Valve', 'Valve')
```

```
insert into jogo ( idJogo,titulo,icone,,publicadora,estudio) values (12, 'Elden Ring',  
'url_ImagemMarioOdyssey', 'Bandai', 'FromSoftware')
```

```
insert into jogo ( idJogo,titulo,icone,,publicadora,estudio) values (13, 'Mario Odyssey',  
'url_ImagemMarioOdyssey', 'url_ImagemMarioOdyssey', 'Nintendo', 'Nintendo')
```

# Esquema Físico e Carga Inicial do Banco de Dados Não-Convencional

Para o banco de dados NoSQL, será utilizado o MongoDB, que cria as coleções de acordo com a inserção de um documento nelas pelo método insert. O MongoDB funciona por meio de coleções, que por sua vez armazenam documentos que guardam pares de chaves-valor.

Como as coleções são dinâmicas e não há delimitação dos atributos que os documentos dentro de uma coleção precisam ter, ao contrário dos bancos relacionais que delimita o esquema, e os atributos de cada linha em suas tabelas, o esquema físico do banco não-relacional será sua própria carga inicial.

Coleção JogosPossuídos:

```
jogo_possuido_1 = {
  uuid: 'ac6bce24-6e0f-46d6-80ea-b09d7214356a',
  idUsuário: 1,
  idJogo: 12 ,
  plataforma: 'PS5',
  data_adicao: '2024-05-15',
  estado: 'nao_jogado'
};
jogo_possuido_2 = {
  uuid: 'f6c25ac9-a5bd-f758-f612-a59f2f920323',
  idUsuário: 2,
  idJogo: 11 ,
  plataforma: 'PC',
  data_adicao: '2024-04-03',
  estado: 'jogando',
  feedback: 'O FPS mais clássico de todos',
  horas_jogadas: 200.6
  nota: 10
};
jogo_possuido_3 = {
  uuid: '1eb1145e-120a-ac07-3087-40068bbb9242',
  idUsuário: 3,
  idJogo: 13 ,
  plataforma: 'Switch',
  data_adicao: '2024-05-12',
  estado: 'finalizado',
  feedback: 'Incrível, o Mario 3D que mais joguei',
  horas_jogadas: 150
  nota: 8
};
db.jogos.insertMany([jogo_possuido_1,jogo_possuido_2,jogo_possuido_3]);
```



# Diagrama do Banco de Dados Não-Convencional

Coleção JogosPossuidos
<ul style="list-style-type: none"><li>- uuid : 'f6c25ac9-a5bd-f758-f612-a59f2f920323'</li><li>- idUsuário : 2</li><li>- idJogo : 11</li><li>- Plataforma : 'PC'</li><li>- data_adicao : '2024-04-03'</li><li>- estado : 'jogando'</li><li>- feedback : 'O FPS mais clássico de todos'</li><li>- horas_jogadas : 200.6</li><li>- nota : 10</li></ul>